

AMENDMENTS

Amendments to the Title:

Please replace the title (shown on the cover page and pages 1 and 27) with the following rewritten title:

INTELLIGENT SYSTEM CONTROL AGENT FOR MANAGING JOBS ON A NETWORK

Amendments to the Specification:

Please replace the paragraph beginning on page 3, line 26, with the following rewritten paragraph:

A similar problem may arise if the job request submitted to the selected client is too complex for the client to execute. For instance, a job may be too complex if it entails computation that would be computationally prohibitively expensive or slow to finish or ~~a~~if it requires resources such as RAM or disk space which ~~exceeds the resource~~ ~~exceed the resources~~ installed on the client. This may cause a client to become overloaded and terminate and/or lose other job requests located in the client's queue.

Please replace the paragraph beginning on page 5, line 11, with the following rewritten paragraph:

In one embodiment, the intelligent system control agent also comprises a ~~state awareness module~~ ~~system health check module~~ configured to be aware of the state of the client. An agent communication protocol module may be used to communicate with the software located within the client. The intelligent system control agent may also provide in one embodiment an agent endpoint module configured to allow the mobility of an agent from one system to another. A

federation module may be used to allow cross-communication and interaction between multiple agents. A job relocation module may also be used to relocate a user requested job from one client to another. A state storage module may be provided to store the state of jobs being relocated from one client to another.

Please replace the paragraph beginning on page 6, line 25, with the following rewritten paragraph:

In one embodiment, an agent communication protocol module is configured to communicate with the software located within the client. The method may comprise configuring a ~~state awareness module~~ system health check module to be aware of the state of the client. The method may also comprise using a federation module to cross-communicate and interact between multiple agents.

Please replace the paragraph beginning on page 8, line 9, with the following rewritten paragraph:

Figure 1 Figure 1 is a schematic block diagram illustrating one embodiment of a computer system, for implementing the intelligent software control agent system of the present invention.

Please replace the paragraph beginning on page 11, line 24, with the following rewritten paragraph:

In the depicted embodiment of Figure 2, the agent 200 generally comprises a user interface module 203, an agent endpoint module 204, a client selection module 205, a storage mechanism module 206, a federation module 208, a communication module 210, a state storage module 212, a system health check module 214, a registration module 216, and various other communication channel modules 218, and a job relocation module 219.

Please replace the paragraph beginning on page 12, line 1, with the following rewritten paragraph:

In the depicted embodiment of Figure 2, the client 202 generally comprises job execution queues 220, a job execution module 221, a queue-monitoring module 222, a communication module 224, a status module 226, and various other communication channel modules 228. Additionally, the client 202 may be coupled to a stub software module 229.

Please replace the paragraph beginning on page 13, line 24, with the following rewritten paragraph:

In one embodiment, the client modules 202 comprise job execution queues 220 and a job execution module 221. The job execution module 221 may determine which type of job execution queue might be used to process a particular user request. The job execution queues 220 may comprise various types of job execution queues. In the depicted embodiment, the job execution queues 220 comprise an asynchronous queue, a synchronous queue, and an exclusive queue. A job execution queue 220 may also be configured to maintain several types of each queue previously mentioned. However, a job execution queue 220 is configured to maintain a maximum of one exclusive queue for each client. Figure 3 shows one example in which a job execution queue 220 comprises all three types of queues.

Please replace the paragraph beginning on page 14, line 5, with the following rewritten paragraph:

Preferably, the queue monitoring module 222 is configured to periodically check each job execution queue 220 and to execute the jobs contained therein, subject to criteria specified in the job, queue, and client metadata. In certain embodiments where the job to be executed comprises off-the-shelf software or software in which hooks cannot be previously added to the software since the

source code is not available, a stub module (not shown) 229, residing outside the agent 200 and client 202, is used to provide basic job control, such as “stop, start, restart, cancel, etc.” on behalf of the monitoring module 222. As mentioned before, a stub module 229 typically provides the capability of conducting basic control functions, such as stop, start, restart, cancel, etc. In these embodiments, the stub module 229 does not provide robust job management functions such as state management. The use of the stub module 229 is intended to allow the agent 200 to control software that was not created with distributed focus in mind. Software that is created to take advantage of the agent’s features is typically provided with its own job management function, but may also be provided with additional features for robust job management.

Please replace the paragraph beginning on page 14, line 19, with the following rewritten paragraph:

As used herein, the term “metadata” is intended to include any type of data which describes other data. Metadata is preferably stored for all objects-object types in the system, including clients, agents, jobs, and domains.

Please replace the paragraph beginning on page 16, line 1, with the following rewritten paragraph:

Figure 4 is a schematic flow chart diagram illustrating one embodiment of a process 400 for receiving, sending, and executing user requests in accordance with the present invention. In one embodiment, the process 400 will be described in conjunction with the system of figures-Figures 1 through 3, but it should be understood that the process 400 may also be conducted independent of that system. The process 400 starts 402, and an agent 200 comes online 404 with a network. The agent 200 proceeds to determine 406 if relocation is necessary. If an agent 200 is required to be relocated, the process 400 continues to the steps 408 that are illustrated in Figure 5 and which will be explained hereafter.